ALGORITMOS Y ESTRUCTURAS DE DATOS III
Final / 20-JUL-2021

1. [2.5 puntos] Dado un conjunto de números reales $S = \{x_1, x_2, \ldots, x_n\}$, se quiere determinar un conjunto de menor cardinalidad de intervalos cerrados de longitud 1 que contenga a todos los números de S. Diseñar un algoritmo goloso que resuelva este problema.

   Por ejemplo, para $S = \{-1.5, -1, 2\}$, resultados posibles son:

   - $\{[-2, -1], [1, 2]\}$
   - $\{[-1.5, -0.5], [1.5, 2.5]\}$

   Mientras que $\{[-2.1, -1.1], [-1, 0], [1.1, 2.1]\}$ no cumple tener la mínima cardinalidad posible.

   Demostrar que el algoritmo propuesto es correcto y determinar su complejidad temporal. Justificar.

2. [2.5 puntos] Un corte de aristas de G es un conjunto de aristas tal que al borrarlas aumenta la cantidad de componentes conexas de G. Dado $G = (V, X)$ un grafo conexo, probar que un subgrafo H de G es subgrafo de $G' = (V, X \setminus X_T)$ para algún $T = (V, X_T)$ árbol generador de G si, y sólo si, H no contiene un corte de aristas de G.

3. [2.5 puntos] Dado un flujo máximo que puede circular por una red, donde c_e es la capacidad máxima de la arista e y f_e es el valor del flujo en el arco e, decimos que un arco es **vital máximo** si al eliminarlo de la red se produce el máximo decrecimiento del valor del flujo (obtenido eleminando sólo un arco).

   Decir si son verdaderas o falsas las siguientes afirmaciones. Justificar.

   (a) Un arco **vital máximo** es un arco e que tiene el valor máximo de c_e.
   (b) Un arco **vital máximo** es un arco e que tiene el máximo valor de f_e.
   (c) Un arco **vital máximo** es un arco e que tiene el máximo valor de f_e, entre los que pertenecen a un corte mínimo.
   (d) Un arco que no pertenece a un corte de capacidad mínima no puede ser un arco **vital máximo**.
   (e) Una red puede contener varios arcos **vitales máximos**.

4. [2.5 puntos] En el problema SUBSET-SUM la entrada es un conjunto $S = \{x_1, \ldots, x_n\}$ y un entero t. En COMPOSITE, dado un entero n, se quiere saber si n es un número compuesto (es decir, n tiene un factor p, con $1 < p < n$). Sabiendo que SUBSET-SUM es NP-completo y COMPOSITE es NP, decidir si cada una de las siguientes sentencias se desprenden de estos hechos (justificar todas las respuestas).

   (a) SUBSET-SUM $\leq_p$ COMPOSITE.
   (b) Si hay un algoritmo $O(nt)$ para SUBSET-SUM, then P = NP.
   (c) Si hay un algoritmo $O(n^3 \log t)$ para SUBSET-SUM, entonces hay un algoritmo polinomial para COMPOSITE.
   (d) Si hay un algoritmo $O(\log n)$ para COMPOSITE, entonces P = NP.
   (e) Si P $\neq$ NP, entonces ningún problema en NP puede ser resuelto en tiempo polinomial.